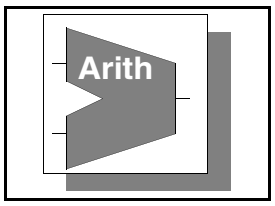
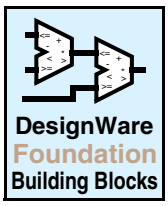




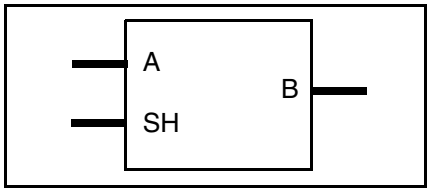
DW01_bsh

Barrel Shifter

Version, STAR, and myDesignWare Subscriptions: IP Directory

Features and Benefits

- Parameterized data and shift coefficient word lengths
- Inferable using a function call



Revision History

Description

DW01_bsh is a general-purpose barrel shifter. Shifted data wraps around from the MSB to the LSB.

The recommended value of the parameter *SH_width* is related to *A_width* by the equation:

$$SH\_width = ceil(\log_2[A\_width])$$

If all shift combinations are not needed, reduce *SH_width* to less than $ceil(\log_2[A\_width])$ to save hardware. *SH_width* can be larger than $ceil(\log_2[A\_width])$, but that will generate redundant hardware.

Table 1-1 Pin Description

Pin Name	Width	Direction	Function
A	*A_width*	Input	Input data
SH	*SH_width*	Input	Shift control
B	*A_width*	Output	Shifted data out

Table 1-2 Parameter Description

Parameter	Values	Description
A_width	≥ 2 (See Table 1-4 on page 2)	Word length of `A` and `B`
SH_width	≤ ceil($\log_2$[*A_width*])	Word length of `SH`

Table 1-3 Synthesis Implementations[a]

Implementation Name	Function	License Feature Required
str	Synthesis model target for speed	DesignWare
astr	Synthesis model target for area	DesignWare

a. During synthesis, Design Compiler will select the appropriate architecture for your constraints. However, you may force Design Compiler to use any architectures described in this table. For more, see *DesignWare Building Block IP User Guide*.

The following table lists example values for *A_width* and the corresponding *SH_width* values. For example, if *A_width* = 8, *SH_width* = 3 because there can be at most $2^3 = 8$ shift combinations.

Table 1-4 Sample Parameter Values

A_width	*SH_width*
2	1
3 - 4	2
5 - 8	3
9 - 16	4
17 - 32	5
33 - 64	6

Table 1-5 Simulation Models

Model	Function
DW01.DW01_BSH_CFG_SIM	Design unit name for VHDL simulation
dw/dw01/src/DW01_bsh_sim.vhd	VHDL simulation model source code
dw/sim_ver/DW01_bsh.v	Verilog simulation model source code

Table 1-6 Truth Table (*A_width*=3)

SH(2:0)	B(7)	B(6)	B(5)	B(4)	B(3)	B(2)	B(1)	B(0)
000	A(7)	A(6)	A(5)	A(4)	A(3)	A(2)	A(1)	A(0)
001	A(6)	A(5)	A(4)	A(3)	A(2)	A(1)	A(0)	A(7)
010	A(5)	A(4)	A(3)	A(2)	A(1)	A(0)	A(7)	A(6)
011	A(4)	A(3)	A(2)	A(1)	A(0)	A(7)	A(6)	A(5)
100	A(3)	A(2)	A(1)	A(0)	A(7)	A(6)	A(5)	A(4)
101	A(2)	A(1)	A(0)	A(7)	A(6)	A(5)	A(4)	A(3)
110	A(1)	A(0)	A(7)	A(6)	A(5)	A(4)	A(3)	A(2)
111	A(0)	A(7)	A(6)	A(5)	A(4)	A(3)	A(2)	A(1)

Related Topics

- Logic – Combinational Overview
- DesignWare Building Block IP User Guide

HDL Usage Through Function Inferencing - VHDL

```
library IEEE, DWARE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;
use DWARE.DW_foundation_arith.all;

entity DW01_bsh_func is
  generic(func_A_width : integer:=8; func_SH_width : integer :=3);
  port(func_A: in std_logic_vector(func_A_width-1 downto 0);
       func_SH: in std_logic_vector(func_SH_width-1 downto 0);
       B_func_TC: out std_logic_vector(func_A_width-1 downto 0);
       B_func_UNS: out std_logic_vector(func_A_width-1 downto 0));
end DW01_bsh_func;

architecture func of DW01_bsh_func is
begin
  B_func_TC <= std_logic_vector(DWF_bsh(SIGNED(func_A),
                                UNSIGNED(func_SH)));
  B_func_UNS <= std_logic_vector(DWF_bsh(UNSIGNED(func_A),
                                UNSIGNED(func_SH)));
end func;
```

HDL Usage Through Function Inferencing - Verilog

```
module DW01_bsh_func (func_A, func_SH, B_func);
  parameter func_A_width = 8;
  parameter func_SH_width = 3;
    // Passes the widths to the bsh function
  parameter A_width = func_A_width;
  parameter SH_width = func_SH_width;

  // Please add search_path = search_path + {synopsys_root + "/dw/sim_ver"}
  // to your .synopsys_dc.setup file (for synthesis) and add
  // +incdir+$SYNOPSYS/dw/sim_ver+ to your verilog simulator command line
  // (for simulation).
  `include "DW01_bsh_function.inc"

  input [func_A_width-1:0] func_A;
  input [func_SH_width-1:0] func_SH;

  output [func_A_width-1:0] B_func;

  assign B_func = DWF_bsh(func_A, func_SH);
endmodule
```

HDL Usage Through Component Instantiation - VHDL

```
library IEEE,DWARE;
use IEEE.std_logic_1164.all;
use DWARE.DWpackages.all;
use DWARE.DW_foundation_comp.all;

entity DW01_bsh_inst is
  generic (inst_A_width : POSITIVE := 8;
           inst_SH_width : POSITIVE := 3);
  port (inst_A : in std_logic_vector(inst_A_width-1 downto 0);
        inst_SH : in std_logic_vector(inst_SH_width-1 downto 0);
        B_inst : out std_logic_vector(inst_A_width-1 downto 0));
end DW01_bsh_inst;

architecture inst of DW01_bsh_inst is
begin

  -- Instance of DW01_bsh
  U1 : DW01_bsh
    generic map ( A_width => inst_A_width, SH_width => inst_SH_width )
    port map ( A => inst_A, SH => inst_SH, B => B_inst );
end inst;

-- pragma translate_off
configuration DW01_bsh_inst_cfg_inst of DW01_bsh_inst is
  for inst
  end for; -- inst
end DW01_bsh_inst_cfg_inst;
-- pragma translate_on
```

HDL Usage Through Component Instantiation - Verilog

```
module DW01_bsh_inst( inst_A, inst_SH, B_inst );

  parameter A_width = 8;
  parameter SH_width = 3;

  input [A_width-1 : 0] inst_A;
  input [SH_width-1 : 0] inst_SH;
  output [A_width-1 : 0] B_inst;

    // Instance of DW01_bsh
  DW01_bsh #(A_width, SH_width)
    U1 ( .A(inst_A), .SH(inst_SH), .B(B_inst) );

endmodule
```

Revision History

For notes about this release, see the *DesignWare Building Block IP Release Notes*.

For lists of both known and fixed issues for this component, refer to the STAR report.

For a version of this datasheet with visible change bars, click here.

Date	Release	Updates
January 2019	DWBB_201806.5	■ Updated example in "HDL Usage Through Component Instantiation - VHDL" on page 5 ■ Added this Revision History table and the document links on this page

Copyright Notice and Proprietary Information

Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Trademarks

Synopsys and certain Synopsys product names are trademarks of Synopsys, as set forth at https://www.synopsys.com/company/legal/trademarks-brands.html.

All other product or company names may be trademarks of their respective owners.

Free and Open-Source Software Licensing Notices

If applicable, Free and Open-Source Software (FOSS) licensing notices are available in the product installation.

Third-Party Links

Any links to third-party websites included in this document are for your convenience only. Synopsys does not endorse and is not responsible for such websites and their practices, including privacy practices, availability, and content.

Synopsys, Inc.
www.synopsys.com